# Лабораторная работа №3 Servlets. Java Server Pages

### Темы для предварительного изучения

Протокол HTTP

HTML

CSS

XML

JAXB

Servlets

JSP

### Задание на лабораторную работу

В процессе разработки учебного приложения ознакомиться с созданием Web-приложений с применением Java-технологий.

### Задание 1

Разработать html-страницы для ввода данных и для вывода результата.

Первая страница должна называться index.html. На ней должны находиться:

* заголовок,
* сопроводительный текст,
* два поля для ввода данных для выполнения операции,
* приглашения (подсказки) к этим полям ввода,
* кнопка для завершения ввода (с переходом на result.html).

Вторая страница должна называться result.html. На ней должны находиться:

* заголовок,
* значения аргументов,
* результат выполнения операции,
* ссылка на исходную страницу,
* ссылка на скачивание файла result.xml.

На этапе разработки дизайна в качестве значений аргументов и результата выполнения операции могут использоваться произвольные данные.

Обе страницы должны быть оформлены с применением CSS. Таблица стилей должна быть размещена в отдельном файле и быть общей для обеих страниц. Все элементы обеих страниц должны изменить свой внешний вид в соответствии со стилями (дизайн – на ваше усмотрение). Особое замечание: ссылки должны менять цвет при наведении на них курсора.

### Задание 2

На основе страницы index.html следует разработать страницу index.jsp, добавляющую следующую функциональность.

1. Форма должна переходить на эту же страницу.

2. Если запрос содержит введённые данные, то эти данные должны быть проверены на корректность в соответствии с бизнес-логикой действия, определяемого вариантом (см. таблицу 1).

3. Если данные некорректны, то они должны быть снова выведены в форму, а некорректный элемент выделен и снабжён комментарием (выделение также средствами CSS).

4.Если данные корректны, они должны быть помещены в объект класса (его также нужно разработать), описывающий задание на выполнение. Объект должен быть помещён в сессию как атрибут, после чего обработка запроса должна быть передана на адрес result.html.

### Задание 3

К адресу result.html должен быть привязан сервлет, который выполняет следующие действия.

1. Извлекает из сессии объект задания.

2. Выполняет задание, помещая результат выполнения в тот же объект.

3. Передаёт обработку запроса на страницу result.jsp.

### Задание 4

На основе страницы result.html следует разработать страницу result.jsp, добавляющую следующую функциональность.

1. Задание и результат его выполнения извлекаются из сессии, значения используются для вывода на страницу.

2. Если объект задания в сессии отсутствует, следует вывести сообщение об ошибке и перенаправить запрос на index.jsp с задержкой в 5 секунд и прямой ссылкой для немедленного перехода.

### Задание 5

К адресу result.xml должен быть привязан сервлет, выполняющий следующие действия.

1. Тип отклика изменяется на xml.

2. Задание и результат извлекаются из сессии.

3. На их основе формируется отклик следующего вида.

--- result.xml --------------------------------------------------------------------------------------------------

<task>

<argument1>Argument value</argument1>

<argument2>Argument value</argument2>

<result>Result value</result>

</task>

-------------------------------------------------------------------------------------------------------------------

Для формирования xml-отклика должна использоваться технология JAXB. Класс, используемый для передачи значений в рамках сессии, должен быть соответствующим образом подготовлен.

Таблица 1. Операции для различных вариантов.

|  |  |  |
| --- | --- | --- |
| № варианта | Выполняемое действие | Некорректные данные |
| 1 | Конкатенация строк | Пустые строки |
| 2 | Поиск вхождения второй строки в первую (результат – позиция вхождения) | Пустые строки |
| 3 | Размножение строки (первый аргумент – строка, второй – число, результат – строка, повторенная заданное число раз) | Пустая строка, введено не число, отрицательное число. |
| 4 | Удаление концевых символов (первый аргумент – строка, второй – количество символов, которые надо удалить, результат – уменьшенная строка) | Пустая строка, введено не число, отрицательное число, количество символом больше длины строки. |
| 5 | Удаление начальных символов (первый аргумент – строка, второй – количество символов, которые надо удалить, результат – уменьшенная строка) | Пустая строка, введено не число, отрицательное число, количество символом больше длины строки. |
| 6 | Удаление подстрок (первый аргумент – строка, второй – подстрока, результат – исходная строка, в которой удалены все элементы, совпадающие с подстрокой) | Пустые строки |
| 7 | Сложение двух чисел | Введено не число |
| 8 | Разность двух чисел | Введено не число |
| 9 | Умножение двух чисел | Введено не число |
| 10 | Частное двух чисел | Введено не число |
| 11 | Возведение в степень | Введено не число, первое число отрицательное. |
| 12 | Логарифм первого числа по основанию второго | Введено не число, первое число отрицательное. |